09 587 826

TITLE OF THE INVENTION

METHODS AND SYSTEMS FOR AUTOMATED
INFORMATION RETRIEVAL

5

PRIORITY APPLICATION

This application claims the benefit of US Provisional Application No. 60/138,348

filed June 9, 1999, entitled "System and Method for Automated Information Retrieval

(AIRS)," and is incorporated herein by reference.

10

CROSS REFERENCE TO RELATED APPLICATIONS

This application is related to US Application Serial No. 09/551930, filed April 19,

2000, entitled "Platform-Independent Exceptions-Based Methods and Systems for

Remotely Managing Nodes Within a Communications Network," and is incorporated

15   herein by reference.

COPYRIGHT NOTIFICATION

BACKGROUND OF THE INVENTION

1.   Field of the Invention

The present invention relates to computerized information retrieval systems and, more particularly, to automated methods and systems for selecting, retrieving, storing, and managing transaction logs and other data, such as, for example, data from automated teller machines, financial institution servers, and almost any network node device.

2.   Background

Transaction logs are records of customer activity on a particular Automated Teller Machine (ATM) or a particular network node device. They are used for a number of important purposes. One important purpose, for example, is for problem resolution. If a customer has a dispute about a transaction that took place at an ATM, the transaction log provides a record of at least one side of the issue. Another important purpose, for example, is to track overall activity at an ATM or for a set of ATMs to understand customer usage patterns. Still, another important purpose, is to facilitate compliance with government reporting regulations.

The current process that businesses, such as financial institutions, undergo in collecting ATM transaction logs and other data is manual and requires tremendous manpower and coordination efforts. This problem is not restricted to a particular financial institution but is an industry-wide problem. Thus, there is a current need for an automated mechanism for selecting, retrieving, storing, and managing transaction logs

and other data, which would provide a tremendous savings in terms of both time and

money on an industry-wide basis.

SUMMARY OF THE INVENTION

To overcome these problems, the present invention provides easy, reliable, and

5   efficient methods and systems for automatically selecting, retrieving, storing, and

managing transaction logs and other data, such as, for example, data from automated

teller machines, financial institution servers, and almost any network node device.

In a preferred embodiment of the present invention, the methods and systems can

be configured to automatically select, retrieve, store, and manage various types of files,

10   with each type of file having its own frequency of upload, upload schedule, destination

directory, and the like. In an another embodiment, the methods and systems can be

configured to automatically select, retrieve, store, and manage various types of files for

nodes that have missed days (i.e., days when the node is down, such as when the node is

out of service) of data and that have other missing data. In an another embodiment, the

15   methods and systems can be configured to prioritize the selection, retrieval, storage, and

management of various types of files, including files for nodes that have missed days or

that have other missing data.

In an embodiment of the methods and systems of the present invention, a network

provides a front end interface, such as, for example, in the form of a computer software

20   application that provides a template for viewing, selecting, and entering information in

order to select, retrieve, store, and manage transaction logs and other data. For example,

instead of manually gathering ATM transaction logs at an ATM site, log retrievals are

automatically selected, retrieved, and stored in an electronic transfer medium.

In an embodiment of the present invention, the automated information retrieval system is an interactive Graphical User Interface (GUI) and service application projected upon a client terminal running X Windows coupled with a network management system server that is connected to a network. As will be appreciated by those in the art, the invention could also be implemented using a variety of hardware platforms and operating systems, such as, for example, a JAVA program capable of running on any platform that has a JAVA Runtime Environment (JRE), such as, JRE version 2.0.

In an embodiment of the present invention, the user modules represent screens displayed on a client terminal and allow a user to view, input, select, and/or transmit data. For example, the user module known as the "Status" Module allows a user to monitor network node devices and to display the retrieval status for various file types (e.g., transaction log, etc.) of the selected network node device. The retrieval status is refreshed periodically based on the configuration value parameters assigned by an Administrator. A user can also force an immediate status refresh by clicking on a button at the bottom of the screen.

In addition to providing the methods and systems outlined above, the present invention: (1) operates on a multitude of hardware platforms and/or operating systems; (2) provides real-time retrieval of transaction logs and other data; (3) utilizes a user-friendly interactive user interface; (4) provides highly configurable retrieval commands; (5) facilitates automated commands to the network node devices; (6) facilitates automated responses from the network node devices to the network management system

server; and (7) utilizes the network to reach the nodes instead of specialized lines.

In different possible embodiments of the present invention, the methods and systems may be utilized to perform one or more of the following tasks: (1) provide integration with external systems; (2) provide integration with other internal systems; (3)

5   utilize internally maintained data; (4) provide on-line network node device data; (5) allow for real-time system modifications and system configuration; (6) provide detailed reports; and/or (7) utilize state-of-the-art communications technology (e.g., web-based technology).

In a first illustrative embodiment, a platform-independent method for retrieving

10   and managing data in at least one communications network having a plurality of destination nodes interconnected with communication lines, comprises:

remotely accessing a communications network;

remotely configuring a retrieval command associated with a destination node;

15   remotely transmitting said retrieval command to said destination node;

remotely monitoring said retrieval command associated with said destination node;

remotely transmitting a response from said destination node to said retrieval command;

20   remotely monitoring said response from said destination node to said retrieval command; and

remotely storing said response from said destination node to said retrieval

command.

In this first illustrative embodiment, the method may further comprise:

remotely prioritizing said retrieval command associated with said

destination node;

5   remotely prioritizing said response from said destination node to said

retrieval command;

remotely executing an automated retrieval schedule;

remotely constructing a response log;

remotely administering said response log; and

10   remotely printing said response log.

In a second illustrative embodiment, a platform-independent system for retrieving

and managing data in at least one communications network having a plurality of

destination nodes interconnected with communication lines, comprises:

means for remotely accessing said communications network;

15   means for remotely configuring a retrieval command associated with a

destination node;

means for remotely transmitting said retrieval command to said destination

node;

means for remotely monitoring said retrieval command associated with said

20   destination node;

means for remotely transmitting a response from said destination node to

said retrieval command;

means for remotely monitoring said response from said destination node to said retrieval command; and

means for remotely storing said response from said destination node to said retrieval command.

5   In this second illustrative embodiment, the system may further comprise:

means for remotely prioritizing said retrieval command associated with said destination node;

means for remotely prioritizing said response from said destination node to said retrieval command;

10   means for remotely executing an automated retrieval schedule;

means for remotely constructing a response log;

means for remotely administering said response log; and

means for remotely printing said response log.

In a third illustrative embodiment, a method for selecting, prioritizing, retrieving,

15   storing, and managing data within network nodes, comprises:

configuring a request from a user to a network node;

transmitting said request to said network node;

processing said request associated with said network node;

transmitting a response from said network node to said request;

20   processing said response from said network node to said request; and

storing said response from said network node to said request.

In this third illustrative embodiment, the method may further comprise:

constructing a response log;

administering said response log;

printing said response log;

prioritizing said request associated with said network node;

5　　prioritizing said response from said network node to said request;

constructing an automated retrieval schedule;

executing said automated retrieval schedule; and

managing said response associated with said network node.

In a fourth illustrative embodiment, a system for selecting, prioritizing, retrieving,

10　storing, and managing data within network nodes, comprises:

means for configuring a request from a user to a network node;

means for transmitting said request to said network node;

means for processing said request associated with said network node;

means for transmitting a response from said network node to said request;

15　　means for processing said response from said network node to said request;

and

means for storing said response from said network node to said request.

In this fourth illustrative embodiment, the system may further comprise:

means for constructing a response log;

20　　means for administering said response log;

means for printing said response log;

means for prioritizing said request associated with said network node;

means for prioritizing said response from said network node to said request;

means for constructing an automated retrieval schedule;

means for executing said automated retrieval schedule; and

means for managing said response associated with said network node.

5   In the first, second, third, and fourth illustrative embodiments, the present

invention may include an automated mechanism to indicate the retrieval status of data

from a node. This retrieval status indicates whether a retrieval (1) was never attempted,

(2) was successful, (3) is not available, (4) is out of range (e.g., date is out of

configuration parameters), (5) failed, and/or (6) in progress. The automated mechanism

10   to indicate the retrieval status may be configured to retrieve data from nodes using (1) an

upload frequency, (2) an upload schedule, and/or (3) a destination directory.

In these four illustrative embodiments, configuration of the retrieval command

may include parameters for (1) minimum time to retry if retrieval failure, (2) maximum

number of simultaneous retrievals, (3) file type, (4) file type name, and (5) archive

15   directory. Configuration of the retrieval command may also include parameters for (1) at

least one selected day, (2) at least one selected hour, (3) at least one selected said

destination node, (4) at least one missed day, (5) at least one missed hour, (6) at least one

disconnected destination node, (7) at least one down destination node, and (8) at least one

exception-reported destination node.

20   Further, in these four illustrative embodiments, a user may enable node filtering

and select particular node types, nodes affiliated with a particular business, or nodes

affiliated with a particular business branch.

Methods and Systems for Automated Information Retrieval

In a fifth illustrative embodiment, a platform-independent system for retrieving

and managing data in at least one communications network having a plurality of

destination nodes interconnected with communication lines, comprises:

a network automated information retrieval system coupled to at least one

5　communications network having a plurality of nodes;

an interactive user module coupled with a network management system

server connected to said communications network having a plurality of nodes;

a plurality of client terminals coupled to said interactive user module for

user interaction with said network automated information retrieval system.

10　In this fifth illustrative embodiment, the interactive user module may be

communicated by a service application of said automated information retrieval system to

said network management system server. Alternatively, it could be communicated to one

of a internet, an intranet, or an extranet.

In this fifth illustrative embodiment, the system further comprises a request that is

15　communicated to the automated information retrieval system by user interaction with an

interactive user module. The request represents a retrieval command to query at least one

destination node. Additionally, the interactive user module may comprise (1) an

administrator module, (2) an operator module, (3) a help module, and/or (4) a status

module.

20　In addition to the components identified above, this fifth illustrative embodiment

comprises memory, at least one database stored in memory, and at least one database

processor capable of processing data contained in said database. Additional components

may include:

means for said plurality of network nodes to transmit a response to said request.

means for processing said response from said plurality of network nodes to

5   said request;

means for storing said response from said plurality of network nodes to said request;

means for constructing a response log, wherein said response log comprises a plurality of responses from said plurality of network nodes to said request;

10   means for administering said response log; and

means for printing said response log.

Further, in all of the illustrative embodiments above, the nodes (e.g., destination nodes, network nodes, nodes, etc.) may comprise one or more of the following:

a plurality of delivery system nodes;

15   a plurality of secondary system nodes;

automated teller machines;

bank servers;

communication servers; and

financial servers.

20   Finally, in these illustrative embodiments, the communications network may be a financial institution's communications network, such as a bank's communication network.

Further details on these embodiments, other possible embodiments, and the methods and systems of the present invention are set forth below.

As will be appreciated by those of ordinary skill in the art, the methods and systems of the present invention have wide utility in a number of areas as illustrated by

5   the wide variety of features and advantages discussed below.

It is a feature and advantage of the present invention to provide an automated selection, retrieval, storage, and management methods and systems which is generic and platform-independent and can be used, for example, with ATMs, financial institution servers, and almost any network node device.

10   It is another feature and advantage of the present invention to provide automated information retrieval methods and systems which can be configured to select, retrieve, store, compare, and report various types of files for nodes that have missed days of data and that have other missing data.

It is another feature and advantage of the present invention to provide automated

15   information retrieval methods and systems which can be configured to prioritize the selection, retrieval, storage, and management of various types of files, including files for nodes that have missed days or that have other missing data.

It is another feature and advantage of the present invention to identify data that is present on a network node device prior to its entry into the retrieval system.

20   It is another feature and advantage of the present invention to provide flexible configurable operation parameters.

Methods and Systems for Automated Information Retrieval

It is another feature and advantage of the present invention to provide automated information retrieval methods and systems that can be configured to retrieve various types of files, with each type of file having, for example, its own frequency of upload, upload schedule, and destination directory.

5   It is another feature and advantage of the present invention to provide real-time configuration of data retrieval and real-time review of all data, including transaction logs.

It is another feature and advantage of the present invention to automatically retrieve files from a network node device based on a scheduling configuration.

It is another feature and advantage of the present invention to provide scheduling

10   configuration based on: (1) a single selected day for a retrieval period; (2) number of days in retrieval period; and/or (3) hours and/or days of the week in a retrieval period.

It is another feature and advantage of the present invention to automatically retrieve files from a network node device based on a node identification configuration.

It is another feature and advantage of the present invention to automatically

15   retrieve files from a network node device using an exception-based (i.e., error-based) configuration.

It is another feature and advantage of the present invention to provide detailed node configuration based on: (1) missed days in a retrieval period; and/or (2) nodes that were disconnected, down, reported an exception, and the like.

20   It is another feature and advantage of the present invention to provide data filtering.

It is another feature and advantage of the present invention to allow for node

filtering based on one or more selected nodes, nodes affiliated with a particular business, and/or nodes affiliated with a particular business branch (i.e., segments or offices within a business).

It is another feature and advantage of the present invention to provide an

5   opportunity for centralized information processing.

It is another feature and advantage of the present invention to provide detailed logging (e.g., detailed exception-based logging) at the network management system server.

It is another feature and advantage of the present invention to employ data

10   compression to minimize transfer times.

It is another feature and advantage of the present invention to provide for unattended operation.

It is another feature and advantage of the present invention to provide security and entitlement features.

15   It is another feature and advantage of the present invention to secure the executable code running on the network management system server.

It is another feature and advantage of the present invention to provide for multiple levels of user access (e.g., an administrator level and an operator level) and to facilitate multiple levels of security related to those levels of user access.

20   It is another feature and advantage of the present invention to provide network systems functionality to the administrator to ensure timely customer service response and compliance with various legal requirements.

It is another feature and advantage of the present invention to be transport independent.

It is another feature and advantage of the present invention to provide a flexible automated information retrieval system that is capable of accommodating changes in the system architecture.

It is another feature and advantage of the present invention to support new file types to be retrieved.

It is another feature and advantage of the present invention to make it easier for financial institutions to accommodate growth in the number of network nodes.

It is another feature and advantage of the present invention to support new node devices as they are added to and entered into the network.

It is another feature and advantage of the present invention to perform multiple simultaneous uploads for multiple file types in a network of thousands of managed network nodes.

It is another feature and advantage of the present invention to provide automated information retrieval system that is capable of running on many different hardware platforms and with many different operating systems.

It is another feature and advantage of the present invention to interface and communicate with the network communications system through a variety of electronic mediums, including wireline and wireless technology, such as, for example, WAN, LAN, satellite system, telephone lines, and the like.

It is another feature and advantage of the present invention to provide real-time

status.

It is another feature and advantage of the present invention to provide real-time

batch status.

It is another feature and advantage of the present invention to provide extensive

5   exception handling and fail-over facilities.

It is another feature and advantage of the present invention to utilize state-of-the-

art technology and minimize impact on the embedded code running inside the node

resources and to utilize the communications network to reach the nodes instead of

specialized lines in a reliable and efficient manner.

10   It is another feature and advantage of the present invention to provide automated

methods and systems for selecting, retrieving, storing, and managing data from network

nodes, thereby, reducing manpower and coordination efforts.

It is another feature and advantage of the present invention to provide labor

savings by eliminating the need for a person to visit the site of a network node device to

15   extract, download, and store transaction logs and other data.

It is another feature and advantage of the present invention to eliminate

interruptions of network node devices.

It is another feature and advantage of the present invention to significantly reduce

the time required by the overall node management process (i.e., the time it takes to select,

20   retrieve, store, and manage data on network nodes).

It is another feature and advantage of the present invention to minimize the

amount of paper work generated by: (1) selecting, retrieving, storing, and managing data;

(2) configuring data retrieval commands; and/or (3) tracking data.

It is another feature and advantage of the present invention to provide automated information retrieval system for a financial institution's communications network.

It is another feature and advantage of the present invention to provide on-line system help to a user.

It is another feature and advantage of the present invention to allow a user to store data on a local computer or local network.

It is another feature and advantage of the present invention to provide detailed reporting.

It is another feature and advantage of the present invention to have one standardized user interface regardless of a user's computer system (i.e., the hardware, operating system, and/or other software).

It is another feature and advantage of the present invention to provide for the use of similar user modules when one or more users and/or one or more financial institutions are discussing a network node device, file type, and/or other parameters.

These advantages and features may be accomplished singularly, or in combination, in one or more of the embodiments of the present invention.

Additional uses, objects, advantages, and novel features of the invention will be set forth in the detailed description that follows and will become more apparent to those skilled in the art upon examination of the following or upon learning by practice of the invention.

BRIEF DESCRIPTION OF THE FIGURES

Other objects and advantages of the invention will be more clearly understood by reference to the following description taken in connection with the accompanying figures, in which:

Figure 1 illustrates an overview of the network topology in an embodiment of the

5　methods and systems for automatically retrieving information from network nodes.

Figure 2 illustrates a client-server overview in an embodiment of the methods and systems for automatically retrieving information from network nodes.

Figure 3 illustrates a overview of the GUI in an embodiment of the methods and systems for automatically retrieving information from network nodes.

10　Figure 4 illustrates an "Administrator - Status" Module that an administrator views, inputs, selects, and/or transmits status information in an embodiment of the methods and systems for automatically retrieving information from network nodes.

Figure 5 illustrates an "Administrator - Global Configuration" Module that an administrator views, inputs, selects, and/or transmits configuration information in an

15　embodiment of the methods and systems for automatically retrieving information from network nodes.

Figure 6 illustrates an "Administrator - Retrievable File Type" Module that an administrator views, inputs, selects, and/or transmits retrievable file type information in an embodiment of the methods and systems for automatically retrieving information from

20　network nodes.

Methods and Systems for Automated Information Retrieval

Figure 7 illustrates an "Operator - Global Configuration" Module that an operator views, inputs, selects, and/or transmits configuration information in an embodiment of the methods and systems for automatically retrieving information from network nodes.

Figure 8 illustrates an "Operator - Retrievable File Type" Module that an operator

5   views, inputs, selects, and/or transmits retrievable file type information in an embodiment of the methods and systems for automatically retrieving information from network nodes.

Figure 9 illustrates a "User's Guide" Module that provides an on-line user's guide and help mechanism in an embodiment of the methods and systems for automatically retrieving information from network nodes.

10

DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

Referring now in detail to an embodiment of the present invention, the methods and systems for automated information retrieval is designed to retrieve any kind of data or files, and not simply, transaction and/or management information systems (MIS) logs.

15   The automated system for an embodiment of the present invention recognizes and retrieves data that has not yet been centrally collected and archived. The system employs compression to minimize transfer times and is configurable in operational parameters.

The methods and systems in an embodiment of the present invention provides back-up solutions in addition to retrieval of information. Features of the system of the

20   present invention include, for example, a level of automation employing intelligence in portability and in prioritizing data retrieval for nodes with exceptions, failed retrieval attempts, down-time, and the like. The system of the present invention provides

automated retrieval of various types of files or other records from managed network

nodes. Further, the system can be configured to retrieve several types of files, with each

type having, for example, its own frequency of upload, upload schedule, and destination

directory.

5   The system automatically collects data of interest around the network on a priority

basis, such as, data that is missed due to communication downtime or other conditions

and catches up later per a schedule which is part of the system. The system of the present

invention is highly configurable with regard, for example, to the number of days in the

retrieval period, to the hours and days of the week during which information is retrieved,

10   and to selecting certain nodes for data retrieval. For example, the system can be

configured to retrieve information only at night or to retrieve information only from

certain network nodes which are of interest.

An embodiment of the present invention provides, for example, detailed logging

(e.g., exception-based logging data) at the network management system server. The data

15   collected by the system is accessible to users through various means. The system can also

be used, for example, to provide a user another application, such as, retrieving security

violations or occurrences of security violations from certain ATMs or other nodes and

identifying and displaying them to administrators. The user can tailor the kind of search

the user wishes to make to a certain extent. The system for an embodiment of the present

20   invention is extendible, such that it can be configured on the network management system

server side with minimal effort to retrieve all sorts of information. With regard to

searching, utilities are deployed and reside on the remote nodes, and those utilities are

Methods and Systems for Automated Information Retrieval

programmed for a particular task. For example, utilities are provided for retrieving

information, such as, security violations.

In an embodiment of the present invention, detailed exception-based logging is

provided at the network management system server. The system provides entitlement

5   awareness for multiple levels of access, such as, for example, an administrator or an

operator level.

The system of the present invention supports, for example, data compression,

filtering, encryption, and other security features. The system provides up-to-the minute

status displays, and different views are available through column sorting. Batch status

10   reports are available, and the system supports entry of new node types into the network, as

well as new filter types to be required.

The present invention will now be described in more detail by illustrative

examples with reference to the embodiment(s) depicted in the Figures. The following

described embodiment(s) is presented by way of example and should not be construed as

15   limiting the inventive concept to any particular configuration.

Referring to Figure 1, a basic overview of the network topology according to an

embodiment of the present invention is depicted. As shown in Figure 1, the network

management system server **1** is coupled to a plurality of network node device **3a-3f, 4a,**

and **4b** via a network **2** using X.25 or Transmission Control Protocol/Internet Protocol

20   (TCP/IP). X.25 and TCP/IP are basic communication languages or protocols used by

world wide communications networks, such as, the Internet. X.25 and TCP/IP can also be

used as a communications protocol in the private networks called intranets and in

Methods and Systems for Automated Information Retrieval

extranets.

The automated information retrieval system includes at least one client terminal **5a, 5b**. The client terminal **5a, 5b** typically includes a central processing unit (CPU), a monitor or other visual display device, a keyboard or some other input device, and a communications device, such as a modem. Further, each client terminal **5a, 5b** is electronically connected to at least one network **2**.

In an embodiment of the present invention, the client terminal **5a, 5b** may be any PC running a Windows operating system or may be a Windows NT workstation with access to a global communications network, such as, the Internet. For example, the client terminal **5a, 5b** may be a PC that supports X-based Windows as a display platform. Alternatively, it should be appreciated that the client terminal **5a, 5b** could take on a variety of other suitable forms, such as, for example, PC's and/or servers running UNIX or LINUX, a Macintosh, or a pen-based computer. Furthermore, the client terminal **5a, 5b** could be electronically connected to a network management system server **1** by way of other wireline or wireless technology, including, for example, WAN, LAN, satellite system, telephone lines, and the like.

ATMs **3a-3f** transmit and receive data to and from a network management system server **1** via a network **2**. Delivery system servers **4a-4b** also transmit and receive data from the network management system server **1** via the network **2**. Client terminals **5a-5b** interact with the network management system server **1** in a typical client/server platform.

The operation of the system according to the embodiment shown in Figure 1 is as follows. ATM **3a** has transaction log information. For example, transaction log

information may include a balance inquiry, withdrawal, or deposit by a customer to his or

her checking account. As the transaction is processed at ATM **3a**, a record of the

transaction is locally logged and stored at ATM **3a**, and then transmitted to the network

management system server **1** via the network **2**. The network management system server

5   **1** selects, retrieves, and stores transaction record information and other data to the

network **2**.

A user at a client terminal **5a, 5b** enters the system and can (1) observe the

retrieval and storage of transaction record information and other data; (2) control system

configuration (i.e., rules governing selection, retrieval, storage, or management of data);

10   and/or (3) correct system problems. The user can control system configuration based on

the file type, network node device, by business branch, and/or status. Additionally,

network node device data may be selected based on one or more of the following: (1) a

single selected day for a retrieval period; (2) number of days in retrieval period; (3)

hours and/or days of the week in a retrieval period; (4) selected network nodes to retrieve

15   data; (5) missed days in a retrieval period; and/or (6) nodes that were disconnected,

down, reported an exception, and the like.

In an embodiment of the present invention, a node configuration data tool is used

to build and design new nodes to be added to the system. In this manner, the network

management system server **1** and its corresponding client terminals **5a-5b**, may accept

20   and process new network node devices.

In an embodiment of the present invention, not all users are permitted to access the

same system capabilities. Every user of the network management system server **1** must

go through a security check. Users of the network management system server **1** are

organized into a security hierarchy. For example, an administrative user (i.e., an

Administrator) is permitted to access more sensitive system functions and configuration

commands (e.g., adding another type of node or collecting another type of record or files)

5   than an operative user (i.e., an Operator).

A feature of the present invention is the use of universal code, such that the

network management system server **1** can interface with a variety of multiple client

terminals **5a, 5b** to project GUIs and the service application. Furthermore, the service

application enables the network management system server **1** to access the network node

10   devices **3a-3f, 4a**, and **4b** regardless of the platform of the client terminal **5a, 5b**.

In another possible embodiment, the implementation of the present invention may

use a JAVA applet, which may be accessed over a world wide communications network,

such as, for example, the Internet or an Intranet. As will be appreciated by those skilled

in the art, the JAVA language developed by Sun Microsystems of Palo Alto, California, is

15   an example of a language that can be run on a plurality of machines regardless of the

platform of the machine. Thus, in one possible embodiment of the present invention, only

a JAVA application is required for a user to interact with the automated information

retrieval system. Similarly, web-based technology (e.g., web browser and web servers)

could be used to provide user interaction with the automated information retrieval system.

20   Referring now to Figure 2, entitled *"Daemon Components of the Automated*

Information Retrieval System," and Figure 3, entitled *"GUI Architecture of the Automated*

Information Retrieval System," the internal software architecture of the present invention

is illustrated. Many of the boxes represent object classes that are implemented in the

JAVA programming language. Figure 2 illustrates the daemon process that is always

running on the network management system server. Figure 3 illustrates the GUI that is

brought up and down on demand. The present invention is a distributed application

5   wherein a "client" portion **252** runs on any machine with network accessibility to the

corresponding "server," **202** herein referred to as the network management system server.

Figure 2 illustrates the server components **202** and client components **252** of the

distributed software application of the present invention. ACID **204** is the controlling

component for the entire application. It creates the other application components and

10   delegates responsibilities to them in order to accomplish work. It is the conduit through

which the "Cron" component **206** interfaces to the rest of the application.

Cron **206** performs the task of scheduling and managing the components

responsible for the file transfer from network nodes to the server for a specific type of

file. There can be many instances of this component. Each is self-contained, operates

15   against a single node, and is independent of the others. Each manages a single Extraction

script **208**.

Extraction script **208** executes a file extraction script to extract the desired data

from the remote node and puts it into execution. In the network management system

environment, it interacts with the LCserver process **210** (a UNIX-based process that

20   performs remote command execution and remote file transfer). The extraction script **208**

being executed indicates which client-based Extractor **254, 256, 258** to execute, what

characteristics of the data to be extracted are, and where in the server file system that the

LCserver process **210** is to place the uploaded data once it has been retrieved.

Extractor **254, 256, 258** is a client side component that has specific knowledge of

the characteristics of the data that is to be extracted and retrieved. It locates the data,

5   extracts the portion which matches the characteristics specified, and informs the

extraction script of the location from which to retrieve the extraction. There can be

multiple extractors on a client. Each one would be responsible for extracting data from a

different type of data source.

Debug **212** is a development aid for diagnostics and troubleshooting. It is the

10   channel through which the debug information will be displayed.

GUIProxy **214** is responsible for interfacing the GUI component **216** to the rest of

the application. It channels communications between the ACIDProxy component of the

GUI **216** and the ACID component **204**.

ACIDState **218** is the component that is responsible for the bookkeeping as it

15   pertains to which nodes need retrievals and what type of files are to be retrieved. After

each Cron component **206** completes a retrieval (successfully or not), the ACIDState

component **218** is notified of that status and information pertaining to the node that is

updated. Whenever an update occurs, the updated information is immediately written to a

property file **220**. This preserves the state of the file **232** in the event of a server failure.

20   BranchProxy **222** is responsible for the configuration of the application as it

pertains to the nodes of the network from which retrievals will be performed. Over time,

as new nodes are added to the network and others are dropped, BranchProxy **222**

recognizes this and reconciles this information so that the application is always operating

against the proper set of nodes. Node configuration information is supplied to

BranchProxy **222** by the BranchStore service **224**.

Security **226** is responsible for validating the access credentials of any user trying

5   to access the application through the GUI. It verifies that the user is entitled to use the

system and also determines what level of GUI entitlement the user has (i.e., either an

operator or an administrator). In the network management system environment, this is

accomplished by checking against the TS.in (Terminal Security) file **228**. TS.in file **228**

contains the complete list of valid network management system users and their

10 entitlements for each of the network subsystems.

Audit **230** is responsible for adding entries to an audit log. The audit log records

all significant events that occur during the operation of the application, including events

that are user generated and that are system generated.

In an embodiment of the present invention, the AIRS daemon communicates with

15 network nodes (e.g., NT ATMs, etc.) using the LCserver **210** on the network management

system server. The LCserver **210** communicates with managed network nodes using

TTM over TCP/IP. The AIRS daemon communicates with the AIRS GUI **216** and

BranchStore **224** using Remote Method Invocation (RMI). The AIRS daemon is

completely independent of the underlying transport, and it is possible to enable file

20 transfers over HTTP or any other file transfer protocol.

Figure 3 illustrates the GUI Architecture of the automated information retrieval

system **302**. AIRS GUI Dispatcher **304** is the controlling component of the GUI. It

creates the other support components for the GUI and delegates responsibilities to them in order to accomplish work.

ACIDProxy **306** is responsible for interfacing to the AIRS daemon. It channels communications between the AIRS GUI Dispatcher **304** and the GUIProxy **308**

5   component that is part of the daemon component set.

UserNotifier **310** is responsible for displaying exception-based conditions to the user. Whenever a condition occurs that warrants notifying the user that an exception occurred, the UserNotifier **310** component performs that task and then returns the screen to the previous state.

10   MediaStore **312** is responsible for managing graphical images (gif files) **314** on the GUI. These gif files **314** are in Compuserve gif format and reside on the local file system **316**.

AdminScreen **318** is responsible for gathering and displaying the information that composes the AIRS Administration Module. It contains the business logic required to

15   properly manage the screen.

StatusScreen **320** is responsible for gathering and displaying the information that composes the AIRS Status Module. It contains the business logic required to properly manage the module.

DocScreen **322** is responsible for displaying the on-line user documentation that is

20   supplied with the present invention. There are a number of HTML format files **324** that comprise the documentation set.

In an another embodiment of the present invention, configuration information and

the current state of retrievals from the network are maintained in human-readable ASCII

property files, which can be edited by hand in case of unexpected crashed or inability to

bring up the above-mentioned GUI front-end application.

Turning now to the illustrative user modules depicted in Figures 4-9, these user

5   modules represent screen shots that display and prompt the user to view, input, select,

and/or transmit information. For example, some of the information that the user modules

display may include status of the network node devices, types of files and other data,

missing files or missing data, devices that have fallen behind, and/or exceptions. The user

modules may be advantageously displayed as an interactive GUI and service application

10   projected upon a client terminal running X Windows coupled with a network management

system server connected to a network.

Figure 4 depicts an embodiment of the "Status" Module illustrating the retrieval

status for the specific file type. The Status Module is displayed after initialization of the

distributed software application of the present invention and serves as the home module of

15   the AIRS GUI. The retrieval status is refreshed periodically based on the configuration

value in the Refresh Period parameter on the Global Configuration Parameters Tab of the

Administration Screen. A user can also force an immediate status refresh by clicking on

the Refresh Now button at the bottom of the screen.

Detailed descriptions of each GUI components of the Status parameters are

20   described in Table 1 below.

Table 1: Status Parameters

File Type Combo Box	This combo box contains the list of configured file types. A user can view the retrieval status for all the eligible nodes of a selected file type. A user will not be able to see the nodes that do not have a Remote Extractor Path name defined in the Retrievable File Type Parameters Tab of the Administration Module.
Status Table	This table contains the retrieval status for all eligible nodes of a selected file type. The retrieval status is acquired from the NetworkStatus object that is provided by ACIDProxy. A user can also enable or disable retrievals for a specific node or a selection of nodes. To do so, select the node(s) from the Status Table, then click on the Enable button or Disable button. To clear all selections, click on Clear Selection button.
Enable Button	This button will be enabled only if there are nodes selected in the Status Table. If the Enable button is clicked, the button handler retrieves the selected nodes from the Status Table. Then it creates the EnableCommand and invokes method in the ACIDProxy object to update the property file(s) so that the scheduler will attempt to retrieve files/logs for these nodes.
Disable Button	This button will be enabled only if there are nodes selected in the Status Table. If the Disable Button is clicked, the button handler retrieves the selected nodes from the Status Table. Then it creates the EnableCommand object and invokes method in the ACIDProxy object to update the property file(s) so that the scheduler will NOT attempt to retrieve files/logs for these nodes.
Refresh Now Button	Click on this button to force an immediate status refresh to update the status table.
Clear Selections Button	Click on this button to clear all node selections in the Status table.
Report Button	If the Report button is clicked, the button handler invokes method in ACIDProxy to dump the current retrieve status data structure for the selected file type in a readable format in a specified directory on the network management system server.
Status Table Row Height Gage	Adjust the row height in the status table.

In an embodiment of the present invention, the log retrieval results are represented

as follows:

- If retrieval has never been attempted for the specific nodes, the background color of the rows will be painted in light gray.

5

- If retrieval has succeeded for the specific nodes, the background color of the rows will be painted in white.

- If retrieval happened too early and the data was not available at the time for the specific nodes, the background color of the rows will be painted in yellow.

10

- If the date is out of range at the time of the retrieval for the specific nodes, the background color of the rows will be painted in orange.

15

- If retrieval has failed for the specific nodes, the background color of the rows will be painted in red.

- If the system is in the process of retrieving files/logs, the background color of the rows will be painted in green.

20

Figure 5, entitled, "Administration Module - Global Configuration Parameters

Tab" illustrates an embodiment of an administrative module that enables an administrator

to configure the global settings for the daemon and the user interface of the present

invention. When an administrator finishes modifying the global parameters, he or she

25   clicks on the "Apply" icon to set the global parameters.

Detailed descriptions of each GUI component on the Global Configuration

Parameters are described in Table 2 below.

Table 2: Global Configuration Parameters

Enable AIRS Check Box	This allows a user to enable (select) or disable (not select) the AIRS Check Box only if a user has Administrative entitlements. The initial state is acquired from the AdminConfig object that is provided by the ACIDProxy.
Log File Size	This allows a user to configure the file size for the AIRS detail logs only if a user has Administrative entitlements. A valid value is a range from 1 megabyte to 100 megabytes. The initial value is acquired from the AdminConfig object that is provided by the ACIDProxy.
Minimum Retry Time After Retrieval Failure Combo Box	This allows a user to configure the minimum retry time after retrieval failure in hours only if a user has Administrative entitlements. A valid value is a range from 1 to 24 hours. The minimum retrieval retry between hours is acquired from the AdminConfig object that is provided by the ACIDProxy.
Maximum Number of Simultaneous Retrievals Combo Box	This allows a user to configure the maximum number of simultaneous retrievals for the distributed software application of the present invention. A valid value is a range from 1 to "N" where "N" is the maximum possible number of simultaneous retrievals. The maximum number of and possible number of simultaneous retrievals are acquired from the AdminConfig object that is provided by the ACIDProxy.
Enable Tool Tips Check Box	This allows a user to enable (select) or disable (not select) the Tool Tips for all GUI buttons. The initial state is acquired from the AdminConfig object that is provided by the ACIDProxy.
Refresh Period Combo Box	This allows a user to configure the refresh time for the retrieval status on the Status Module. The valid values are 5 seconds, 15 seconds, 30 seconds, 45 seconds, 1 minute, and increments in units of 1 minute afterwards up to 60 minutes. The initial value is acquired from the AdminConfig object that is provided by the ACIDProxy.
Apply Button	If the Apply button is clicked, the button handler retrieves data from all the GUI components on this tab and updates the AdminConfig object using the appropriate set methods. Then it invokes method in the ACIDProxy object to update the property file.
Reset Button	If the Reset button is clicked, the button handler resets the GUI components to present the parameter settings since last Apply.

Figure 6, entitled "Administration Module - Retrievable File Type Parameters

Tab" illustrates an embodiment of an administrative module that enables an administrator

to configure the parameters for each retrievable file type. These parameters are used by

the AIRS daemon to determine whether or not to retrieve files/logs from the network

nodes. If a user has Administrative entitlements, a user can define a new file type by

5   selecting "New" from the File Type Combo Box. If a user chooses to do so, a user must

enter the file type name and specify the retrieval constraints for the new file type. When a

user finishes specifying all the retrieval constraints or making modifications, the user

clicks on Apply.

Detailed descriptions of each GUI component on the Retrieval File Type

10   Parameters are described in Table 3 below.

Table 3: Retrieval File Type Parameters

File Type Combo Box	This combo box contains the list of configured file types. A user can select an existing file type for modifications. If user has Administrator entitlements, the user can modify the archive director, extractor pathnames for the remote nodes of the existing file types, and configure a new file type.
File Type Name Text Field	This is the name of the file type. Normally, this is a read-only field. If a user has Administrator entitlements and selects the "New" icon from the File Type Combo Box, this field will be empty. A user must enter a new file type name.
Retrieve Once Every "N" Day(s) Combo Box	This is the files/logs retrieval frequency in days. A valid value is a range from 1 to 31 days. The initial setting is acquired from the selected FileType object that is provided by the ACIDProxy. If the file type is new, the initial setting is set to 1.
Archive Directory Text Field	The archive directory is the top-level directory for storing the retrieved files/logs on the network management system server. If a user has Administrator entitlements, he or she can modify this field. This archive directory is empty if the file type is new. Otherwise, the archive directory is acquired from the selected FileType object that is provided by the ACIDProxy.
Enable File Type	This allows a user to enable (select) or disable (not select)

Retrieval Check Box	retrieval of files/logs for the selected file type. This field is not selected if the file type is new. Otherwise, the initial setting is acquired from the selected FileType object that is provided by the ACIDProxy.
Initial Catch Up Days Combo Box	This allows a user to configure the initial files/logs retrieval catch up days for the present invention if a user has Administrator entitlements. A valid value is a range from 0 to 31 days. The initial files/logs retrieval catch up days value is acquired from the FileType object that is provided by the ACIDProxy. If the file type is new, the initial setting is set to 0.
Allowable Hours Check Box Table	This allows a user to configure the allowable hours for the files/logs retrieval for the selected file type. The scheduler will attempt to retrieve files/logs based on the constraints specified in this table. All the check boxes in this table are not selected if the file type is new. Otherwise, the initial settings are retrieved from the selected FileType object that is provided by the ACIDProxy.
Remote Node Text Fields (For example: NTC, TCS, etc.) - Remote Extractor Pathname	The number of Remote Extractor Pathname text fields is based on the number of existing node types defined in the branch file on the INC. The names of the node types are used as the labels for each Remote Extractor Pathname text fields. If a user enters the pathname for the specified remote node, the distributed software application of the present invention will attempt to retrieve files/logs from all the remote nodes that have the same target name. If the pathname field is empty, the present invention will not attempt to retrieve logs/files from this type of network node.
Apply Button	If the Apply button is clicked, the button handler retrieves data from all the GUI components on this tab and updates the AdminConfig object using the appropriate set methods. Then it invokes the set method in the ACIDProxy object to update the property file.
Reset Button	If the Reset button is clicked, the button handler resets the GUI components to present the parameter settings since last Apply.

Figure 7 illustrates an "Operator - Global Configuration" Module that illustrates an embodiment of an operator module that enables an operator to configure the global settings for the daemon and the user interface of the present invention. This module is

similar to the "Administration Module - Global Configuration Parameters Tab," except

that some of the configuration options are not available to the operator. Detailed

descriptions of each GUI component on the Global Configuration Parameters are

described in Table 2.

5   Figure 8 illustrates an "Operator - Retrievable File Type" Module illustrates an

embodiment of an operator module that enables an operator to configure the parameters

for each retrievable file type. This module is similar to the "Administration Module -

Retrievable File Type Parameters Tab" except that some of the configuration options may

not be available. Detailed descriptions of each GUI component on the Retrievable File

10   Type Parameters are described in Table 3.

         Figure 9 illustrates a "User's Guide" Module that provides an on-line user's guide

and help mechanism in an embodiment of the methods and systems for automatically

retrieving information from network nodes.

         The foregoing description and associated figures detail only illustrative examples

15   of the environment in which the invention can be used and are not intended to be limiting.

For instance, the user modules shown in Figures 4-9 are by way of example only. In

addition, attributes can be constantly updated and additional fields can be added by

authorized users and/or authorized businesses (e.g., financial institutions). Furthermore,

the programming languages, software platforms, operating systems, hardware

20   components, and other technology mentioned in the foregoing description are by way of

example only, and the present invention may always be enhanced to incorporate the most

advanced available technology. Variations and modifications of the present invention

will be apparent to one skilled in the art, and the above disclosure is intended to cover all

such modifications and equivalents.

GLOSSARY OF ABBREVIATIONS

This disclosure, including illustrative information used to populate user modules in the figures, makes use of certain abbreviations which have the following meanings:

AIRS	Automated Information Retrieval System
ASCII	American Standard Code for Information Interchange
ATM	Automated Teller Machine
CPU	Central Processing Unit
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
INC	Integrated Network Controller
LAN	Local Area Network
LCserver	Local Control Server
RMI	Remote Method Invocation
TCP/IP	Transmission Control Protocol/Internet Protocol
WAN	Wide Area Network